正則パターン上の決定木の学習アルゴリズム (計算機科学の基礎理論 : 21世紀の計算パラダイムを目指して)

寺田, 幹治; 向内, 康人; 佐藤, 優子

数理解析研究所講究録
1148巻 2000年 64-69

正則パターン上の決定木の学習アルゴリズム

寺田 幹治* 向内 康人† 佐藤 優子†
Mikiharu Terada Yasuhito Mukouchi Masako Sato
* 大阪府立大学大学院 理学系研究科
† 大阪府立大学 総合科学部

概 要： パターンとは，定数記号と変数記号からなる文字列である．パターン上の決定木は，内部ノードのラベルがパターンであり，葉のラベルが0または1のラベルをもつ決定木である．寺田他は，各パターンに対して，co-パターンという文字列を導入し，その意味をパターン言語の補言語のある部分集合として定義した．そして，パターン上の決定木の意味を定義し，深さが高々 n である決定木で定義される言語の族が正例から推論可能であることを論じた．本稿では，正の事例から深さが高々2のある特定の形をした正則パターン上の決定木を効率的に学習するアルゴリズムを構築する．

事例からの正則パターン上の決定木の学習問題は，DNA配列からのモチーフの発見というゲノム情報科学の観点からPAC学習の枠組みを用いて研究されてきた．本稿では，この問題を正例からの極限同定という枠組みで論じている．

1 はじめに

パターンとは，アルファベット Σ に含まれる定数記号と変数からなる文字列である．例えば，$p = axby, q = axbx$ は $\{a, b\}$ 上のパターンである．パターン p の意味 $L(p)$ は，p に出現するすべての変数に空でない文字列を代入して得られる Σ 上の文字列からなる言語である．言語を生成するパターンが存在するとき，その言語を**パターン言語**という．各変数が高々1回しか出現しないパターンは**正則パターン**と呼ばれる．**パターン上の決定木** T は，内部ノードのラベルがパターンで，葉のラベルが1または0の決定木である．Σ 上の文字列 w が与えられると，パターンをラベルとするノードでは w がそのパターン言語に含まれるか否かに応じて左または右に分岐し，根から葉へ至る．ただし，w の長さがノードのパターンの長さより小さいならば，そこで停止する．停止することなく，根から葉に至るとき，葉のラベルが1ならば，文字列 w は受理され，0ならば受理されない．決定木 T が定義する言語は，T で受理される Σ 上の文字列の集合である．パターン p に対して，co-パターンと呼ばれる文字列 p^c を導入する．p^c の意味 $L(p^c)$ を $L(p)$ には含まれず，p の長さ以上の文字列の集合と定義すると，上記の決定木で定義される言語は，パターン言語やco-パターン言語に対して，高々有限回の積 (共通部分) 演算や和演算を施して得られる言語である．

パターン言語の族は，正例から帰納推論可能な言語族として Angluin[2] によって導入された．正例からの帰納推論とは，与えられた正の事例からそれらを説明する一般的な規則・表現 (例えば，パターンやパターン上の決定木等) を推測する過程である．本稿では，Gold[4] の極限同定と呼ばれる推論の成功基準による正例からの帰納推論の枠組みを採用し，上記のパターン上の決定木の正例からの帰納推論を考える．

言語演算の観点から推論可能性を扱った研究に，Wright[11] による「有限の弾力性」という概念がある．Wright は，有限の弾力性をもつ言語族が正例から推論可能であり，また，この性質が和演算に関して閉じていることを示した．さらに，パターン言

語族が有限の弾力性をもつこと，したがって，高々 n 個のパターン言語の和からなる言語族も有限の弾力性をもち，正例から推論可能であることを示した．Moriyama&Sato[6] は，有限の弾力性が積演算や連接演算等に関しても閉じていることを示した．演算の閉包性に関するこれらの結果を用いると，もし，パターン言語とそれらの補言語からなる言語族が有限の弾力性をもつならば，深さが高々 d のパターン上の決定木で定義される言語の族は，有限の弾力性を有し，正例から推論可能となる．

寺田他 [10] は，深さが高々 k であるパターン上の決定木で定義される言語族が正例から推論可能であることを示した．また，深さ 1 の正則パターン上の決定木が正例から多項式時間で極限同定可能となる学習アルゴリズムを与えた．本稿では，深さが高々2のある特定の形をした決定木を正例から効率的に学習するアルゴリズムを与える．

パターン上の決定木は，DNA 配列からのモチーフの発見のモデルとして，Arikawa et al.[3] や Miyano[5] 等で採用され，計算学習のパラダイムの 1 つである「PAC 学習」の枠組みを用いて展開されてきた．本稿では，正の事例から正則パターン上の決定木を学習する問題を扱う．

2 パターン上の決定木が生成する言語

2.1 パターン言語

Σ を少なくとも 2 個以上の文字を含むアルファベットとし，X を Σ と互いに素な可算集合とする．Σ, X の要素をそれぞれ**定数** (**記号**)，**変数** (**記号**) とよぶ．

パターンとは，$\Sigma \cup X$ 上の空でない文字列である．パターン p の長さを $|p|$ で表し，パターン全体の集合を $\mathcal{P}$ で表す．**代入**とは，すべての定数をそれ自身に写すパターンからパターンへの準同型写像のこととする．代入 θ によるパターン p の像を $p\theta$ で表す．本稿では，代入として，非消去代入，すなわち，$|x\theta| \geq 1$ $(x \in X)$ を満たす代入 θ に制限して考える．パターン p, q に対して，$p = q\theta$ となる代入 θ が存在するとき，p は q の**例化** (q は p の**汎化**) といい，$p \preceq q$ と表す．明らかに，$p \preceq q$ ならば，$|p| \geq |q|$ である．$p \preceq q$ かつ $q \preceq p$ となるのは，p と q が変数名の付け替えで等しくなる場合に限られる．本稿では，このような 2 つのパターンを同一視することにする．$p \preceq q$ かつ $p \neq q$ のとき，$p \precneqq q$ と記す．パターン p が生成する言語 $L(p)$ を，

$$L(p) = \{w \in \Sigma^+ \mid \exists\theta \text{ s.t. } w = p\theta\}$$

と定義する．例えば，$\Sigma = \{a, b\}$ とするとき，$p = axby$ $(x, y \in X)$ はパターンであり，p が生成する言語は，

$$L(p) = \{aaba, aabb, abba, abbb, aaaba, \cdots\}$$

である．言語 L を生成するパターンが存在するとき，L は**パターン言語**と呼ばれる．定義から，任意の文字列 $w \in \Sigma^+$ に対して，$w \in L(p)$ ならば，$|p| \leq |w|$ となることがわかる．すなわち，$L(p) \subseteq \Sigma^{\geq |p|}$ であり，また，$L(p)$ の最短の文字列の長さは，$|p|$ である．パターン言語の所属問題は，計算可能であるが NP 完全であることが示されている (Angluin[1])．また，$p \preceq q$ ならば，$L(p) \subseteq L(q)$ であることが容易にわかる．しかし，その逆は一般には成り立たない (Angluin[1])．パターン言語の族を $\mathcal{PL} = \{L(p) \mid p \in \mathcal{P}\}$ で表すことにする．

2.2 パターン上の決定木

パターン上の決定木とは，葉の各ノードが 0 または 1 の，葉ではない各ノードがパターンのラベルをもつ決定木である．深さが高々 $n \in N$ であるパターン上の決定木の全体を $\mathcal{TP}_n$ で表し，高々 $n \in N$ 個のパターンを含む決定木の全体を $\mathcal{TP}^n$ で表す．与えられた文字列 $w \in \Sigma^*$ に対して，パターン p をラベルとしてもつノードでは，w がそのパターン言語 $L(p)$ に属するか否かがテストされ，その結果 (yes または no) により左または右に分岐していく．ただし，w の長さがパターン p の長さより小さいならば，分岐せずそこで停止すると仮定する．このテストと分岐は，根と呼ばれるトップのノードで始まり，葉と呼ばれる末端のノードにたどりつくまで行われる．パターン上の決定木 T に対して，途中で停止することなく，1 のラベルをもつ葉のノードに至る文字列の集合を**決定木 T によって定義される言語**といい，$L(T)$ で表す．図 1 の決定木 T は，3 個のパターンをラベルとしてもつ深

さ 2 の決定木である．そして，T によって定義される言語は，

$$L(T) = (L(xaby) \cap L(axbby^c)) \cup (L(xaby^c) \cap L(xayb))$$

である．ただし，パターン p に対して，p^c を p の co-パターンと呼び，その意味を

$$L(p^c) = \{w \in L(p)^c \mid |w| \geq |p|\}$$

と定義する．特に，p が変数だけからなるパターンのときは，$\chi_{|p|}$ 或いは単に χ とかき，χ^c を ϕ とかく．

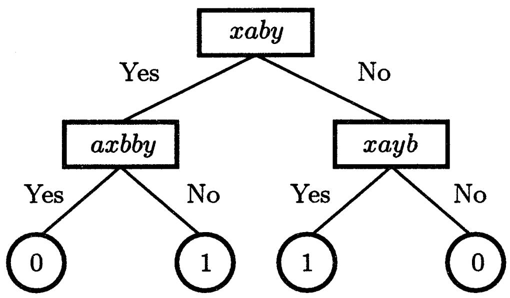

図 1: パターン上の決定木の例

この例でわかるように，パターン $p_1, p_2, \cdots, p_n$ をラベルにもつノードを経由して，途中で停止することなく 1 のラベルをもつ葉のノードで受理される言語は，$L(p_i)$ または $L(p_i^c)$ の i についての共通部分であり，n は決定木の深さを越えない．そして，$L(T)$ は，そのような言語の集合和である．

言語族 $\mathcal{TPL}_n = \{L(T) \mid T \in \mathcal{TP}_n\}$, $\mathcal{TPL}^n = \{L(T) \mid T \in \mathcal{TP}^n\}$ とする．

2.3 正則パターン上の決定木と言語の包含問題

正則パターンとは，各変数が高々1 回しか現れないパターンをいう．例えば，$axby$ は正則であるが，$axbx$ は変数 x が 2 回出現するので正則ではない．正則パターンで定義される言語は**正則パターン言語**と呼ばれる．正則パターンの全体および正則パターン言語の全体をそれぞれ，$\mathcal{RP}$ および $\mathcal{RPL}$ で表すことにする．

正則パターンの co-パターンからなる族を co-$\mathcal{RP}$ とし，$\mathcal{JRP} = \mathcal{RP} \cup$ co-$\mathcal{RP}$ とする．

正則パターン上の決定木で定義される言語について考察する．以下，正則パターンを単に，パターンとかく．

パターン上の決定木 T の根のラベルがパターン p で，p の左，右部分木がそれぞれ T_1, T_2 であるとき，$T = (p, T_1; p^c, T_2)$ とかく．ただし，図 2 左のような部分木をパターン q そのもので表わし，図 2 右のような部分木を co-パターン q^c で表わすことにする．

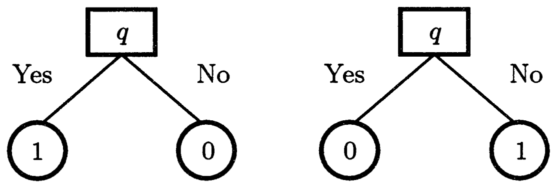

図 2: 高さ 1 のパターン上の決定木

このとき，

$$L(T) = (L(p) \cap L(T_1)) \cup (L(p^c) \cap L(T_2))$$

と表すことができる．また，l_T を言語 $L(T)$ に含まれる文字列の最短の長さとする．

補題 2.1 $T = (p, T_1; p^c, T_2)$ とする．

(i) $|p| \leq l_T$.

(ii) $|p| < l_T$ ならば，$L(T) = L(T')$ かつ，$|p'| = l_T$ となる $T' = (p', T_1'; p'^c, T_2')$ が存在する．

上記の結果から，言語 $L(T)$ を生成する T のノードのラベルはすべて長さが l_T 以上のパターンとしてよい．また，長さ l_T の文字列を受理するので，根からラベルが 1 である葉に至る path の中には，長さが丁度 l_T であるパターンからなる path が存在しなければならない．このような T を**標準形**という．すなわち，T の任意の部分木 T' の根 p の長さは，$l_{T'}$ である．

補題 2.2 T, T' を標準形の決定木とするとき，次の関係が成立する：

$$L(T) \subseteq L(T') \iff L(q) \cap L(T) \subseteq L(T_1'), L(q^c) \cap L(T) \subseteq L(T_2').$$

ただし，$T' = (q, T_1'; q^c, T_2')$ とする．

上記の結果から，深さ k,k' の決定木 T,T' に対する意味的包含問題 $L(T)\subseteq L(T')$ は，深さが高々 $k+1$ と $k'-1$ の決定木で生成される高々2つの包含問題に帰着する．特に，次の補題が成立する．

補題 2.3 τ_1,τ_2 をパターンまたはco-パターンとするとき，次の関係が成立する:

$$L(T)\subseteq L(T')$$
$$\Longleftrightarrow L(q)\cap L(T)\subseteq L(\tau_1), L(q^c)\cap L(T)\subseteq L(\tau_2).$$

ただし，$T'=(q,\tau_1;q^c,\tau_2)$ とする．

従って，意味的包含関係 $L(T)\subseteq L(T')$ を調べるためには，基本的な包含関係 $L(\pi_1)\cap L(\pi_2)\cap\cdots\cap L(\pi_n)\subseteq L(\tau)$ (π_i,τ はパターンまたはco-パターン) か否かを調べればよい．例えば，図1の決定木 T に対して，$L(T)\subseteq(L(q)\cap L(s_1))\cup(L(q^c)\cap L(s_2^c))$ (ただし，$q,s_1,s_2\in\mathcal{RP}$) とすると，この包含問題は，次の連立の基本包含問題に帰着する:

$$L(q)\cap L(xaby)\cap L(axbby^c)\subseteq L(s_1),$$
$$L(q)\cap L(xaby^c)\cap L(xayb)\subseteq L(s_1),$$
$$L(q^c)\cap L(xaby)\cap L(axbby^c)\subseteq L(s_2^c),$$
$$L(q^c)\cap L(xaby^c)\cap L(xayb)\subseteq L(s_2^c).$$

パターンの有限集合 P,Q に対して，次の順序関係を導入する:

$$P\sqsubseteq Q\Longleftrightarrow \forall p\in P,\exists q\in Q \text{ s.t. } p\preceq q.$$

定理 2.4 $\sharp\Sigma\geq 2k-1, q_i\neq\chi(i=1,\cdots,k)$ とする．

(i) $\bigcap_{i=1}^m L(p_i)\cap\bigcap_{j=1}^k L(q_j^c)\subseteq L(q)$
$\Longleftrightarrow \bigcap_{i=1}^m L(p_i)\subseteq L(q)\cup\bigcup_{j=1}^k L(q_j)$
$\Longleftrightarrow \mathrm{mi}(\{p_1,\cdots,p_m\})\sqsubseteq\{q,q_1,\cdots,q_k\}$

(ii) $\bigcap_{i=1}^m L(p_i)\cap\bigcap_{j=1}^k L(q_j^c)\subseteq L(q^c)$
$\Longleftrightarrow L(q)\cap\bigcap_{i=1}^m L(p_i)\subseteq\bigcup_{j=1}^k L(q_j)$
$\Longleftrightarrow \mathrm{mi}(\{q,p_1,\cdots,p_m\})\sqsubseteq\{q_1,\cdots,q_k\}$

ただし，パターン集合 P に対して，$\mathrm{mi}(P)$ は P の極大例化からなる集合とする．

特に，決定木 T,T' の深さが高々 n ならば，$k\leq 2n-1$ に制限される．

この定理は，基本的な意味的包含を構文的な包含に帰着する定理であるが，包含に関する次のCompactness定理を用いて証明することができる．

補題 2.5 (Sato et al.[8], Mukouchi[7]) $\sharp\Sigma\geq 2k-1$ とする．

$$S_1(p)\subseteq L(q_1)\cup\cdots\cup L(q_k)$$
$$\Longleftrightarrow L(p)\subseteq L(q_1)\cup\cdots\cup L(q_k)$$
$$\Longleftrightarrow \exists i\ (1\leq i\leq k) \text{ s.t. } p\preceq q_i$$

ただし，$S_1(p)=\{w\in L(p)\mid |w|=|p|\}$ とする．

定理2.4を上記の例に適用すると，次の4つの関係が得られる:

$$\mathrm{mi}(\{q_1,xaby\})\sqsubseteq\{axbby,q_2\},$$
$$\mathrm{mi}(\{q_1,xayb\})\sqsubseteq\{xaby,q_2\},$$
$$\mathrm{mi}(\{xaby,q_3\})\sqsubseteq\{axbby,q_1\},$$
$$\mathrm{mi}(\{xayb,q_3\})\sqsubseteq\{xaby,q_1\}.$$

また，定理2.4を用いると，深さ2の決定木の言語の包含に関する次の結果が得られる．

補題 2.6 $\sharp\Sigma\geq 5$ とし，$T=(q,\tau_1;q^c,s_2)$, $T'=(p,\pi_1;p^c,\pi_2)$ を標準形の決定木で，$n=l_T=l_{T'}$ かつ，$p,\pi_2,q,s_2\neq\chi,\phi$ を満たすものとする．

$L(T')\subseteq L(T)$ ならば，$\pi_2\in\mathcal{RP}$ である．

3 パターン上の決定木の帰納推論

3.1 言語の帰納推論

文字列の無限列 $w_1,w_2,\cdots$ が言語 L の**正提示**であるとは，$\{w_n\mid n\geq 1\}=L$ が成り立つことである．文字列の無限列 $\sigma=w_1,w_2,\cdots$ の n 番目までの初期部分列を $\sigma[n]$ で表す．

言語族 $\mathcal{L}=L_1,L_2,\cdots$ が**帰納的言語の添字付き族** (indexed family of recursive languages) であるとは，次のような計算可能関数 $f:N\times\Sigma^*\to\{0,1\}$ が存在することをいう: $f(i,w)=1$, if $w\in L_i$; 0, if $w\notin L_i$. ここでの添字 i は言語を定義するオートマトンや形式文法等を意味すると考えられるが，本稿ではパターンやパターン上の決定木等を意味する．

推論機械 M とは，次々に入力を要求し，次々に出力を生成する実行的な手続きのことであり，M が生成する出力を**推測**と呼ぶ．文字列の無限列 σ に対して，その有限列 $\sigma[n]$ が入力された後，M が生成する推測を $M(\sigma[n])$ で表す．推論機械 M が入力の列 σ に対して，添字 $g\in N$ に**収束する**とは，ある $m\in N$ が存在し，任意の $n\geq m$ に対し

て，$M(\sigma[n]) = g$ となることをいう．推論機械 M が言語 L を正例から**極限同定** (identification in the limit) **する** (または，単に**推論する**) とは，L の任意の正提示に対して，M が $L_i = L$ となる添字 i に収束することをいう．また，任意の言語 $L \in \mathcal{L}$ を正例から極限同定する推論機械 M が存在するとき，言語族 $\mathcal{L}$ は正例から**推論可能である**という．

推論機械 M が言語族 $\mathcal{L}$ を正例から推論し，各入力を受け取ってから推測を出力するまでに必要な時間がそれまでの入力の長さの和のある多項式で押えられる時，その推論機械 M は言語族 $\mathcal{L}$ を正例から**多項式時間** (**の更新**) **で推論する**という．また，言語族 $\mathcal{L}$ を正例から多項式時間 (の更新) で推論する推論機械が存在する時，その言語族は，正例から多項式時間 (の更新) で推論可能であるという．

正例からの決定木の帰納推論に関しては，次の結果が得られている．

定理 3.1 (寺田他 [10]) 任意の $n \in N$ に対して，言語族 $\mathcal{TPL}_n, \mathcal{TPL}^n, \mathcal{TRPL}_n, \mathcal{TRPL}^n$ はいずれも正例から推論可能である．

3.2 効率的な帰納推論

上記で述べたように，$\mathcal{TRPL}_n$ 等の言語族は，有限の弾力性 [11] という集合論的性質を有する．パターン上の決定木の族 $\mathcal{T}$ に対して，族が定義する言語族が有限の弾力性を有するならば，次の学習アルゴリズムは，目標決定木 $T \in \mathcal{T}$ を正例から極限同定することが知られている．

```
Algorithm IA
入力: 文字列の無限列 w1, w2, ···;
出力: パターン上の決定木の無限列 T1, T2, ···;
begin
    S := φ;   T0 := (φ, χ; χ, φ);   n := 1;
    repeat
        read the next datum wn;   S := S ∪ {wn};
        if wn ∉ L(Tn-1) then Tn := MINL_T(S)
        else Tn := Tn-1;
        output Tn;   n := n + 1
    forever
end.
```

上のアルゴリズムに使われる手続き $\mathrm{MINL}_{\mathcal{T}}(S)$ はサンプル S を含む極小言語を定義する決定木 $T \in \mathcal{T}$ の1つを計算する手続きである．すなわち，$T = \mathrm{MINL}_{\mathcal{T}}(S)$ ならば，$S \subseteq L(T') \subsetneq L(T)$ を満たす決定木 $T' \in \mathcal{T}$ は存在しない．有限の弾力性は，与えられたサンプル S に対して，S の極小言語が高々有限個であることを保証する．

このアルゴリズムから，(i) 所属問題 $w \in L(T)$ および，(ii) 手続き MINL の計算が共に多項式時間で計算可能ならば，決定木の族 $\mathcal{T}$ は正例から多項式時間で推論可能となる．正則言語の所属問題に関しては，既に多項式時間で計算可能であることが示されている [9]．その結果を用いると，次の結果が直ちに得られる．

補題 3.2 言語族 $\mathcal{TRPL}_n, \mathcal{TRPL}^n$ における所属問題は，多項式時間で計算可能である．

手続き MINL に関しては，次の結果が得られている．

補題 3.3 (Shinohara[9]) 正則パターン言語族 $\mathcal{RPL}$ において，集合 $S \subseteq \Sigma^+$ の極小言語を生成する長さ n の正則パターンを $O(l_{\max}^2 m)$ の時間で計算する手続き $\mathrm{MINL}_{\mathcal{RP}}$ が存在する．ただし，$l_{\max} = \max\{|w| \mid w \in S\}$, $n = \min\{|w| \mid w \in S\}$, $m = \sharp S$ とする．

補題 3.4 (寺田他 [10]) co-正則パターン言語族 co-$\mathcal{RPL}$ において，集合 $S \subseteq \Sigma^+$ の極小言語を生成する長さ n の co-正則パターンを $O(l_{\max}^2 m)$ の時間で計算する手続き co-$\mathrm{MINL}_{\text{co-}\mathcal{RP}}$が存在する．ただし，$l_{\max} = \max\{|w| \mid w \in S\}$, $n = \min\{|w| \mid w \in S\}$, $m = \sharp S$ とする．

上記の結果を用いて，論文 [10] では，深さが高々1である決定木を効率的に学習するアルゴリズムを与え，次の結果を示した．

定理 3.5 (寺田他 [10]) $\sharp\Sigma \geq 3$ とする．深さが1である正則パターン上の決定木の言語族 $\mathcal{TRPL}_1$ は，正例から多項式時間の更新で推論可能である．

4 深さ2の決定木の学習アルゴリズム

本節では，深さが高々2である決定木で生成される言語の正の事例から目標決定木を効率的に学習するアルゴリズムを与える．ただし，本稿では，次の型をもつ決定木に制限する (図 3):

$$T = (q, \chi; q^c, \tau),\ L(T) = L(q) \cup (L(q^c) \cap L(\tau))$$

深さが 1 の決定木は，$q=\phi$ または，$q=\chi$ の特別の場合である．

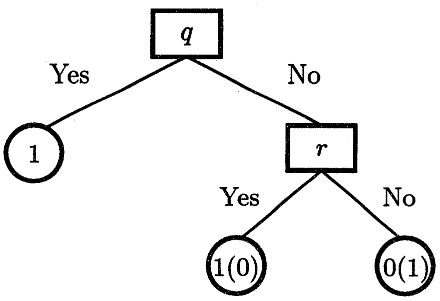

図 3: 高さ 2 の決定木

次の手続き TMINL は，空でない文字列の有限集合 $S \subseteq \Sigma^+$ に対して，上記の制限された決定木の中で S の極小言語を生成する決定木を計算する．

Procedure TMINL(S)
begin
  let n be the length of the shortest strings in S;
  if $\Sigma^n \not\subseteq S$ **then begin**
    $t_0 :=$ MINL(S); $q_0 :=$ co-MINL(S);
    if there is a pair of patterns (t_1, t_2) s.t.
      $t_1 \precsim t_0$, $t_2 \precsim t_0$, $t_1 = \mathrm{MINL}(S - L(t_2))$
      and $t_2 = \mathrm{MINL}(S - L(t_1))$
        then output $T = (t_1, \chi; t_1^c, t_2)$
    else if $t_0 \precsim \chi$ **then output** $T = (t_0, \chi; t_0^c, \phi)$
    else if there is a pair of patterns (q_1, q_2) s.t.
      $q_0 \precsim q_2$, $q_1 = \mathrm{MINL}(S - L(q_2^c))$
      and $q_2^c = \text{co-MINL}(S - L(q_1))$;
        then output $T = (q_1, \chi; q_1^c, q_2^c)$
    else output $T = (\phi, \chi; \chi, q_0^c)$
  end
  else output $T = (\chi, \chi; \phi, \phi)$
end.

このアルゴリズムに関して，次の定理が示される．

定理 4.1 手続き TMINL の出力 T が生成する言語 $L(T)$ は，S の極小言語である．

定理 4.2 深さが高々2 であるパターン上の決定木で生成される言語族は，正例から多項式時間推論可能である．

謝辞

本研究は，文部省科学研究費補助金 特定領域研究 (A)「推論による知識発見に関する研究」(研究課題番号 10143104) の支援を受けました．記して感謝致します．

参考文献

[1] D. Angluin, *Finding patterns common to a set of strings*, in Proceedings of the 11th Annual Symposium on Theory of Computing, (1979) 130–141.

[2] D. Angluin, *Inductive inference of formal languages from positive data*, Information and Control, **45** (1980) 117–135.

[3] S. Arikawa, S. Kuhara, S. Miyano, Y. Mukouchi, A. Shinohara, and T. Shinohara, *A machine discovery from amino acid sequences by decision trees over regular patterns*, New Generation Computing, **11**(3, 4) (1993) 361–375.

[4] E.M. Gold, *Language identification in the limit*, Information and Control, **10** (1967) 447–474.

[5] S. Miyano, *Learning theory towards genome informatics*, in Proceedings of the 4th Workshop on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence, **744** (1993) 19–36.

[6] T. Moriyama and M. Sato, *Properties of language classes with finite elasticity*, IEICE Transactions on Information and Systems, **E78-D**(5) (1995) 532–538.

[7] Y. Mukouchi, *Characterization of pattern languages*, IEICE Transactions on Information and Systems, **E75-D**(4) (1992) 260-267.

[8] M. Sato, Y. Mukouchi and D. Zheng, *Characterisitic sets for unions of regular pattern languages and compactness*, in Proceedings of the 9th International Conference on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence, **1501** (1998) 220–233.

[9] T. Shinohara, *Polynomial time inference of pattern languages and its applications*, in Preceedings of the 7th IBM Symposium on Mathematical Foundations of Computer Science, (1982) 191–209.

[10] 寺田幹治，向内康人，佐藤優子, 正例からのパターン上の決定木の帰納推論, 電子情報通信学会論文誌, **J83-D-I**(1) (2000) 60–67.

[11] K. Wright, *Identification of unions of languages drawn from an identifiable class*, in Proceedings of the 2nd Workshop on Computational Learning Theory, (1989) 328–333.